Integrating MedGemma with LangChain Challenges and Solutions

vertexai, medgemma, gemma-3

zeerakwyne 1 July 31, 2025, 12:01pm

Hi everyone, this is Zeerak from Spryt.

We're working on a multi-agent healthcare system for appointment scheduling and patient support, built using LangChain and LangGraph. Our system uses a sophisticated architecture where different specialized agents (scheduling, rescheduling, cancellation, FAQ) collaborate to handle patient interactions for cervical screening appointments.

Our Setup:

- Multi-agent system using LangChain/LangGraph
- Currently supports Anthropic (Claude), Google (Gemini), and Anthropic Vertex models
- Deployed on Google Cloud using dedicated Vertex AI endpoints
- Production system handling real patient interactions

The Challenge:

We want to integrate MedGemma into our system, but we're facing several technical hurdles:

1. Dedicated Endpoint Issue

We've discovered that LangChain's existing Gemma integration (`GemmaChatVertexAIModelGarden`) doesn't work with dedicated Vertex AI endpoints. The current implementation is designed for the Model Garden but not for custom DNS endpoints like `mg-endpoint-xxxx.europe-west4-xxxx.prediction.vertexai.goog`.

2. System Message Support

We've confirmed that Gemma models don't support system messages - they only accept the chat format with alternating user/model turns. Our agents rely heavily on system prompts to define behavior, context, and response formats.

3. Tool Calling

Our agents use LangChain's tool calling framework extensively. Since MedGemma doesn't have native tool/function calling support, we need to implement a text-based workaround where:

- Tool definitions are injected into prompts
- Tool calls are wrapped in special markers (e.g., ```` ```tool_code``` ````)
- Responses are parsed to extract tool calls and convert them to LangChain's expected format

What We're Looking For:

1. Has anyone successfully integrated MedGemma with LangChain using dedicated Vertex AI endpoints?
2. Are there existing patterns or wrappers for handling tool calling with MedGemma in a LangChain context?
3. Any best practices for converting system messages + tool definitions into MedGemma's chat format while maintaining conversation coherence?

We're planning to build a custom wrapper that:

- Connects to our dedicated endpoint using the OpenAI client approach
- Converts LangChain messages (including system and tool messages) to MedGemma format
- Implements text-based tool calling with reliable parsing
- Maintains compatibility with LangChain's agent framework

Any insights, code examples, or similar experiences would be greatly appreciated! Happy to share our solution once we get it working.

Thanks!

1 Like

---

fmahvar 3 August 2, 2025, 5:42am

Thank you for sharing! Please allow me to get back to you with a better answer.

1 Like

---

zeerakwyne 5 August 18, 2025, 12:25pm

Looking forward to hearing from you @fmahvar

---

fmahvar 7 August 18, 2025, 2:05pm

Hello @zeerakwyne

System instructions should be prepended to the initial user prompt (see **documentation**).

Gemma cookbook has several example notebooks that might be helpful for reference:

- Tool calling: **Gemma function calling**, **Gemma Agentic AI**, **TxGemma agentic demo**
- Minimal example using LangChain with tools: **Gemma LangChain**

Hope this help!

Fereshteh on behalf of HAI-DEF engineering

1 Like

Bitu79 10 August 10, 2026, 6:27am

> fmahvar:
>
> Hello @zeerakwyne
>
> System instructions should be prepended to the initial user prompt (see **documentation**).
>
> Gemma cookbook has several example notebooks that might be helpful for reference:
>
> - Tool calling: **Gemma function calling**, **Gemma Agentic AI**, **TxGemma agentic demo**
> - Minimal example using LangChain with tools: **Gemma LangChain**
>
> Hope this help!
>
> Fereshteh on behalf of HAI-DEF engineering

STOP. This architecture is entirely non-compliant and legally prohibited in a healthcare environment.

Before you write any more custom wrappers or tool-calling hacks, you need to understand that what you are trying to build is a massive regulatory and patient-safety liability.

You are attempting to use experimental, multi-agent LLM loops to handle cervical screening appointments, rescheduling, cancellations, and patient FAQs. In healthcare, this is not a “customer service bot.” These workflows require certified, audited, and strictly regulated medical software systems (EU MDR, HIPAA, or local health authority standards), not generative AI agents.

Here is why the entire approach is fundamentally flawed and prohibited:

• MedGemma is not a certified medical device.
It is an open-weights research model with no clinical validation, no regulatory approval, and it is not legally permitted to operate as a patient-facing communication or appointment-management system.

• Your workarounds are a safety hazard.
Bypassing system messages by injecting them into user prompts, and using text-based regex hacks (`tool_code`) to interpret API actions, is unacceptable in any real healthcare environment.
A single parsing error or hallucinated tool call could cause a patient to miss a critical cancer screening.
Prompt-engineering "hacks" cannot be used for database writes involving patient care.

• Terms of Service violation.
Google's own ToS explicitly prohibit using foundational models as medical devices or for critical clinical purposes without strict regulatory approval.

It is genuinely baffling that a Google representative responded to this with a cookbook link instead of immediately flagging this as a severe violation of safety and compliance standards.

You cannot fix healthcare logistics with LangChain tool-calling wrappers.
You are building a system that puts patient health at risk and exposes your company to catastrophic legal consequences.

This architecture is not technically, legally, or clinically permissible in any real healthcare system.

Bitu79 11 August 10, 2026, 6:37am

> fmahvar:
>
> Hello @zeerakwyne
>
> System instructions should be prepended to the initial user prompt (see **documentation**).

> Gemma cookbook has several example notebooks that might be helpful for reference:
>
> - Tool calling: **Gemma function calling**, **Gemma Agentic AI**, **TxGemma agentic demo**
> - Minimal example using LangChain with tools: **Gemma LangChain**
>
> Hope this help!
>
> Fereshteh on behalf of HAI-DEF engineering

My formal objection to the Google response:
It is deeply concerning that Fereshteh Mahvar (@fmahvar), acting on behalf of Google, responded to this proposal with a 'Gemma cookbook' link and implementation tips instead of clearly indicating that the described approach severely violates healthcare safety and compliance standards.
Providing technical support for a system that attempts to manage medical appointments using regex-based hacks effectively legitimizes a non-compliant and dangerous medical-software architecture.

Why this entire approach is fundamentally flawed and prohibited:

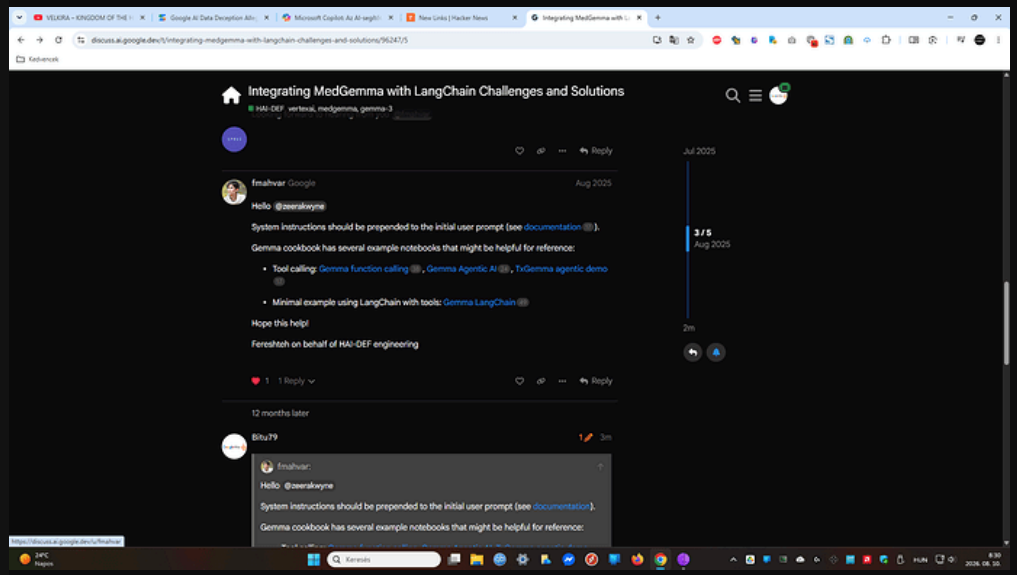
• MedGemma is not a certified medical device.
It is an open-weights research model with no clinical validation, no regulatory approval, and it is not legally permitted to operate as a patient-facing communication or appointment-management system.

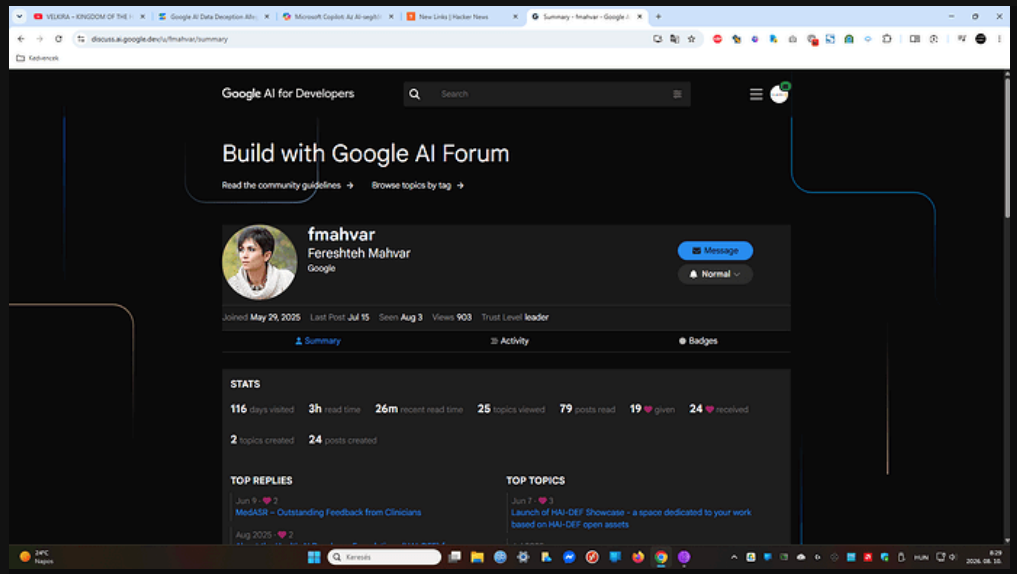
• The described "workarounds" pose a patient-safety risk.
Injecting system instructions into user prompts, using text-based regex tool-calling (`tool_code`), and interpreting API actions through free-form text are unacceptable in any real healthcare environment.
A single parsing error or hallucinated tool call could cause a patient to miss a critical cancer-screening appointment.
Prompt-engineering hacks cannot be used for database operations that affect patient care.

• Violation of Google's Terms of Service.
Google's own ToS explicitly prohibit the use of foundational models for medical or clinical purposes without strict regulatory approval.

Healthcare logistics cannot be solved with LangChain tool-calling wrappers.
A system built this way exposes patients to safety risks and the developer organization to catastrophic legal consequences.

This architecture is not technically, legally, or clinically permissible in any real healthcare system.
I am formally requesting a review of why Google Engineering provided ‘cookbook’-style support for such an obviously non-compliant and regulatorily prohibited use case.


Integrating MedGemma with LangChain Challenges and Solutions
HAI-DEF, vertexai, medgemma, gemma-3
Reply
Jul 2025
fmahvar Google
Aug 2025
Hello @zeerakwyne
System instructions should be prepended to the initial user prompt (see documentation).
Gemma cookbook has several example notebooks that might be helpful for reference:
Tool calling: Gemma function calling , Gemma Agentic AI , TxGemma agentic demo
Minimal example using LangChain with tools: Gemma LangChain
Hope this help!
Fereshteh on behalf of HAI-DEF engineering
3 / 5
Aug 2025
2m
1 Reply
Reply
12 months later
Bitu79
3m
fmahvar:
Hello @zeerakwyne
System instructions should be prepended to the initial user prompt (see documentation).
Gemma cookbook has several example notebooks that might be helpful for reference:



Bitu79 12 August 10, 2026, 6:42am

https://ibb.co/8DNrNycB

https://ibb.co/ksPMkMv8

Related topics

Topic	Replies	Views	Activity
Using MedGemma in Medibound: Launching HIPAA-Ready Agentic Apps in Minutes models, medgemma	1	410	August 18, 2025

Topic	Replies	Views	Activity
Documentation for Developers to Design their Integration with FHIR Systems api, model, medgemma	4	469	July 10, 2025
Gemma 4 27B-A4B-it (MoE) on Vertex AI: vLLM Dependency Triangles, LoRA, and Vision Blockers models, gemma, vertexai, pytorch, transformers	1	381	April 27, 2026
Working environment for running MedGemma (PyTorch/Transformers versions, CUDA, etc.) models, medgemma	2	329	December 14, 2025
MedGemma for Clinical AI Model Selection and Deployment model	2	202	March 26, 2026